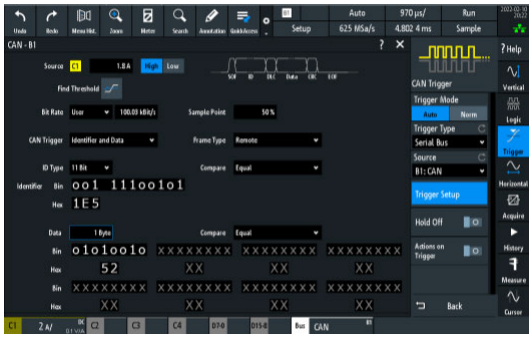
Run the “Protocol” “CAN” app and wire the scope accordingly. Doing it this way, the scope is set completely up for decoding and triggers on new CAN frames.

If you look at the bus table, you should see the simple sequence of 5 frames, sent out every second or so, with messages that form “Rohde & Schwarz RTB2004”. The eight is frame has ID 0x1E5, length 7 bytes, and payload “52 54 42 32 30 30 34” in HEX, which is in ASCII “RTB2004”. (There are some errors reported in earlier frames but that is of no interest now.)

Now try to trigger on that eight frame. Go to CAN trigger setting “Identifier and data”. Set the frame ID on 1E5h. Set the first byte on “52h” and leave the frame length (“Data”) set to 1. Should look like this:



Now the scope does NOT trigger.

Now change increase the frame length (“Data”). Once you reach 7, it starts to trigger. And at 8, it stops triggering again…

PS1: A quick workaround is to set up the data pattern you look for, select the date [length] field, and then turn that slowly up from 1 to 8 bytes. While doing so, *watch the trigger light on the front panel,* the moment it goes on, you found the data length for which such a message exists. A workaround, but not perfect.

PS1: On the positive side, this is a unique way to trigger on frame length (just leave ID and data full of wildcards). But I don’t think it is indented as such, also because the implementation for LIN, SPI, I2C and UART do bot behave this way.